

XML Based Data Management Facility in Support of Frameworks

M. Zubair

Students: D. Tan and A. Jakatdar

Department of Computer Science

Old Dominion University

And

ICASE, NASA Langley

Contact: zubair@cs.odu.edu

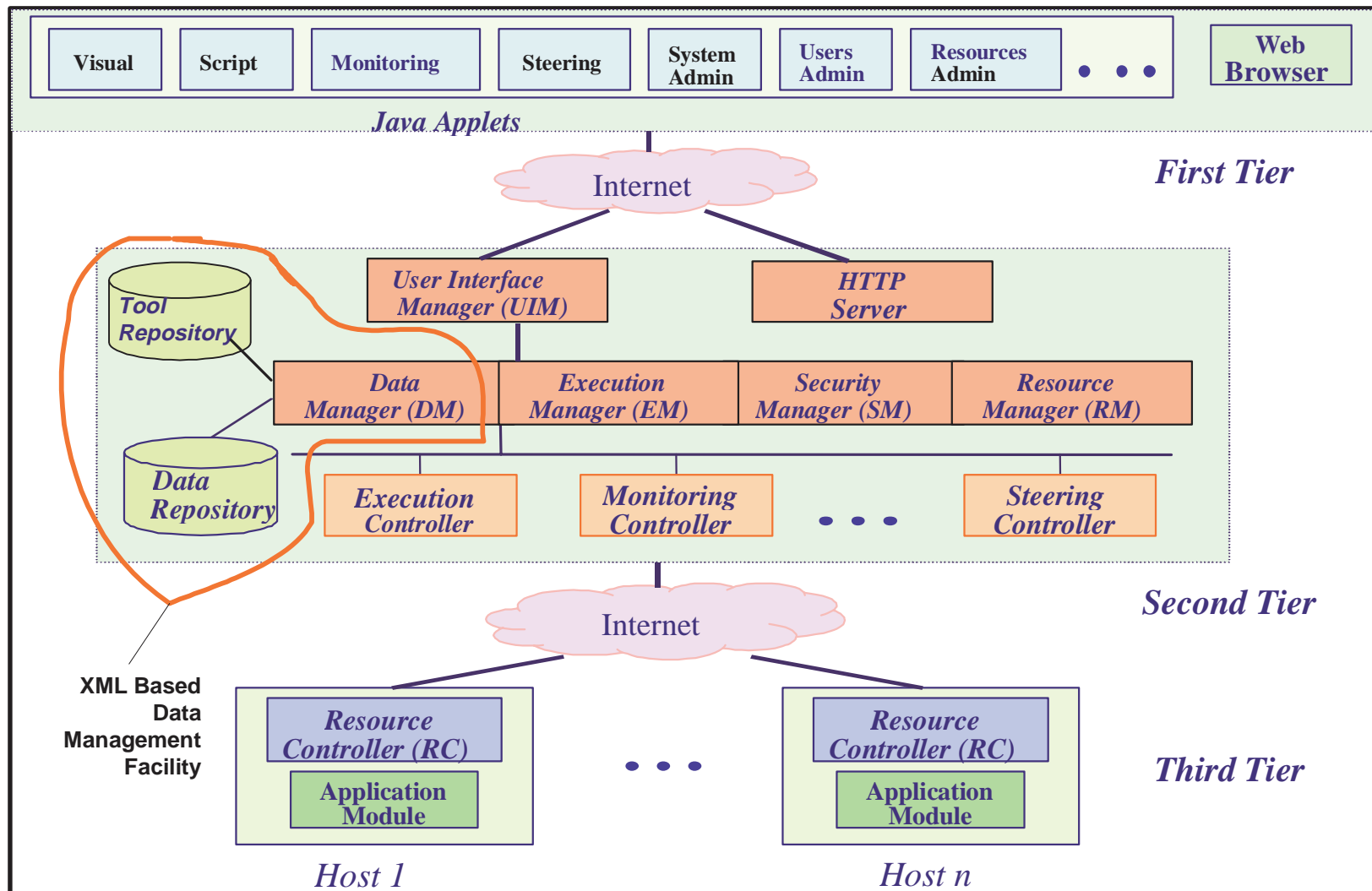
in collaboration with

P. Mehrotra, NASA Ames

HPCCP Workshop on Frameworks, NASA Ames, January 30

Outline

- ☐ Motivation
- ☐ Background
- ☐ Architecture
- ☐ Prototype Demo
- ☐ Future Work



. The ARCADE System Architecture

Motivation

Example Scenario: An application designer is interested in using the wind tunnel data for his code. The wind tunnel data is at a remote site and is not in the desired format, and the transformation tool is at some other remote site.

Lot of scientific data is being generated, experimentally or programmatically, and is not getting shared because of lack of standards and infrastructure support

Often data is consumed by application in a format other than the one in it is produced.

Requirements

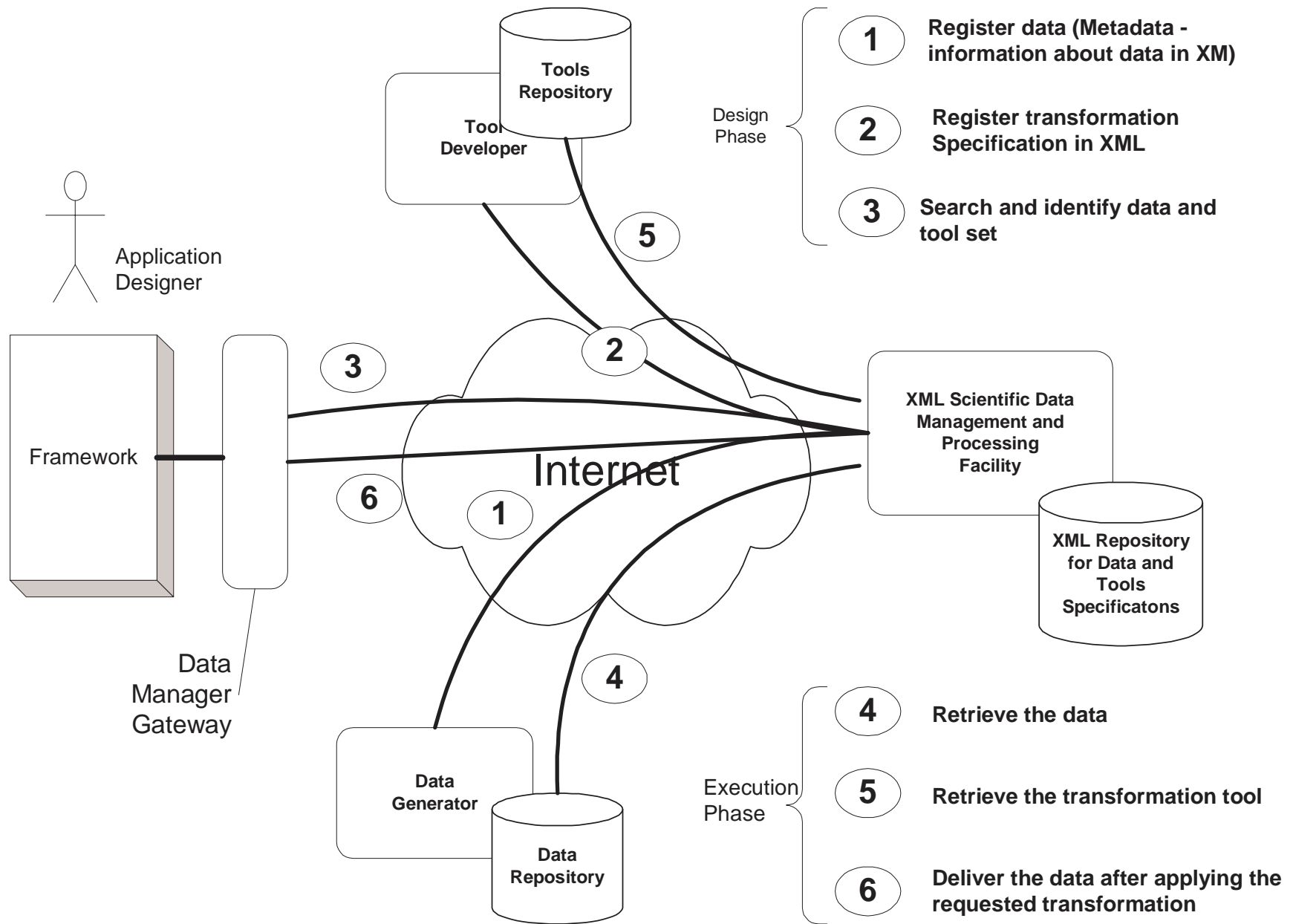
Portable

Encoding data so that it can be moved, transformed, and consumed easily

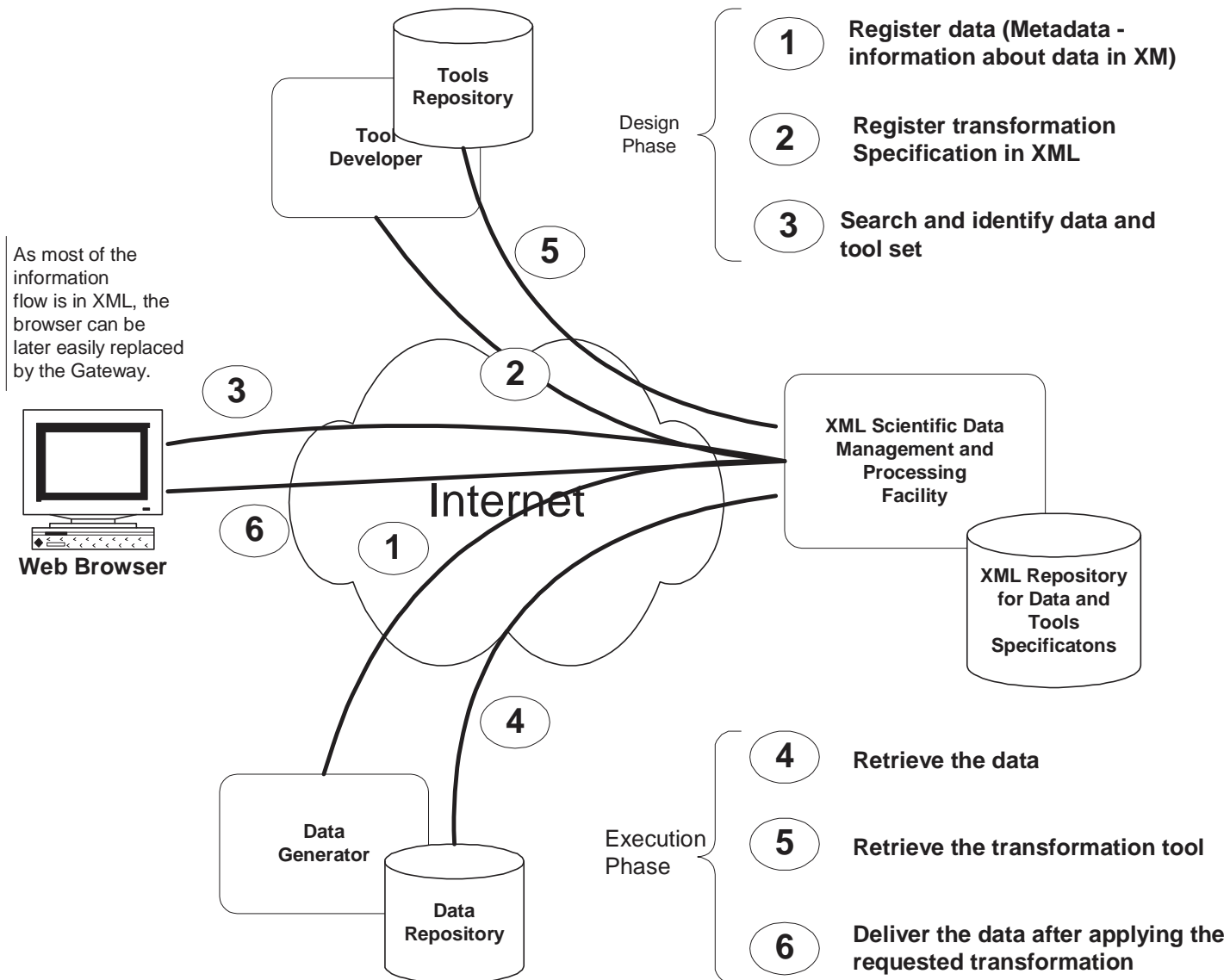
Discoverable

Archiving so that data and associated tools can be retrieved and searched easily

Note: Our current focus is not on the high performance data transfer requirements.



Phase I - Standalone Implementation



Background

XML:

HTML is more suitable for human-computer interaction.

XML helps in separating the structure from presentation, making it suitable for machine-machine interaction.

Standard technology - Lot of tools and packages available from different vendors

Easy to build middleware using XML technology like parsers, stylesheet transformers, etc..

Background

XSLT: Extensible Stylesheet Language Transformations

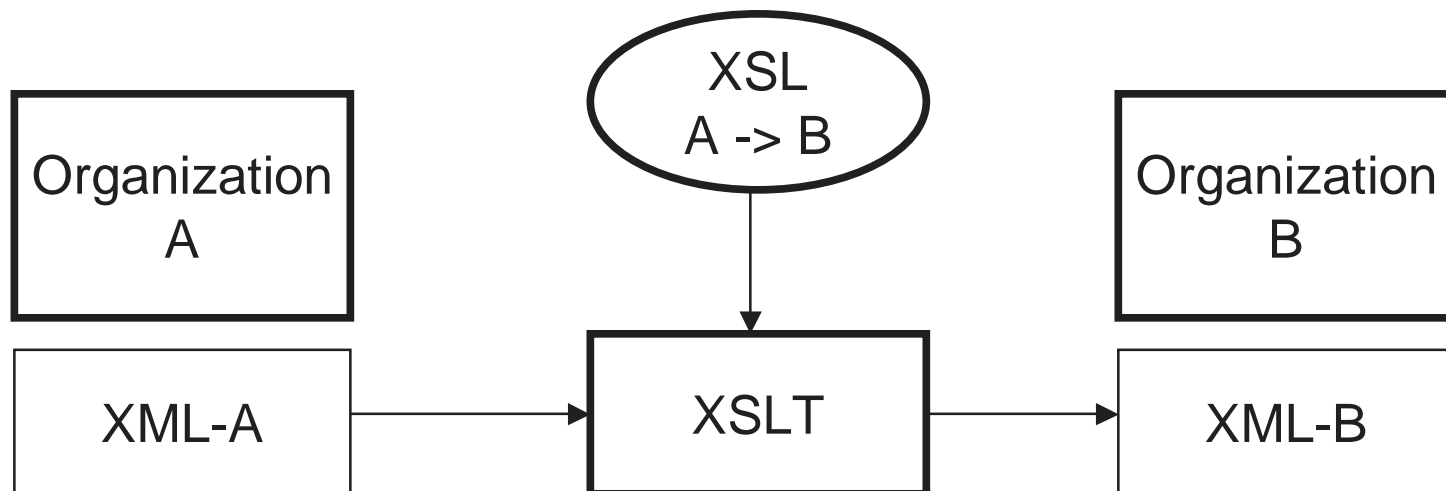
Transforms information using one vocabulary into an alternate form.

Example Implementation: *Xalan* in Java by Apache

XSL: Extensible Stylesheet Language

Describes translations.

Key to Interoperability



Background

XSIL: Extensible Scientific Interchange Language, CACR, Caltech

Support small size data: "Local" (in line)

```
<?xml version="1.0"?>
```

```
<XSIL>
```

```
  <Array Name="Coordinates" Type="float">
```

```
    <Dim>4</Dim>
```

```
    <Dim>2</Dim>
```

```
    <Stream Encoding="Text" Type="Local"
```

```
      Delimiter=", ">1, 0, 1, 1, 0, 1, -1, 1</Stream>
```

```
  </Array>
```

```
</XSIL>
```

Background

XSIL...

Support large size data: "Remote"

```
<?xml version="1.0"?>
<XSIL>
  <Array Name="Coordinates" Type="float">
    <Dim>4</Dim>
    <Dim>2</Dim>
    <Stream Type="Remote" Delimiter=", ">data.dat</Stream>
  </Array>
</XSIL>
```

Very large XML files are not efficient -- most of the XML software can not handle large files.

Background

Scientific Data Management and Digital Library

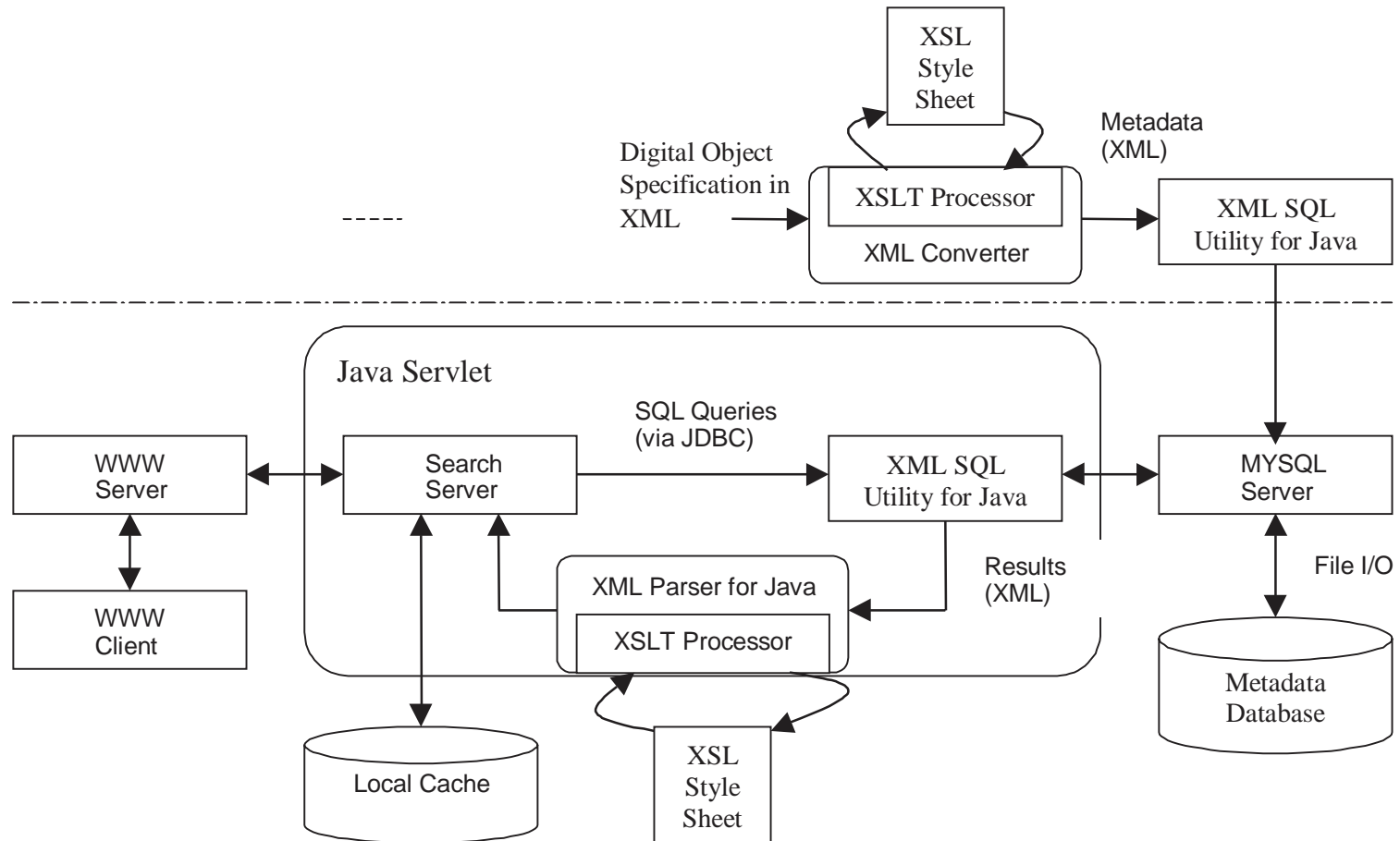
Need to store information about scientific data, and transformation tools such that these can be discovered and retrieved easily.

Digital Library Group At Old Dominion University

- Auto-Classification and Integration of Digital Repositories (Sponsor: United States Joint Forces Command)
- NCSTRL+ -- A bucket based digital library (Sponsor: NASA Langley Research Center)
- Undergraduate Digital Library Framework (Sponsor: NSF)
- Technical Report Interchange Project (Sponsor: LANL, NASA LaRC, AFRL)

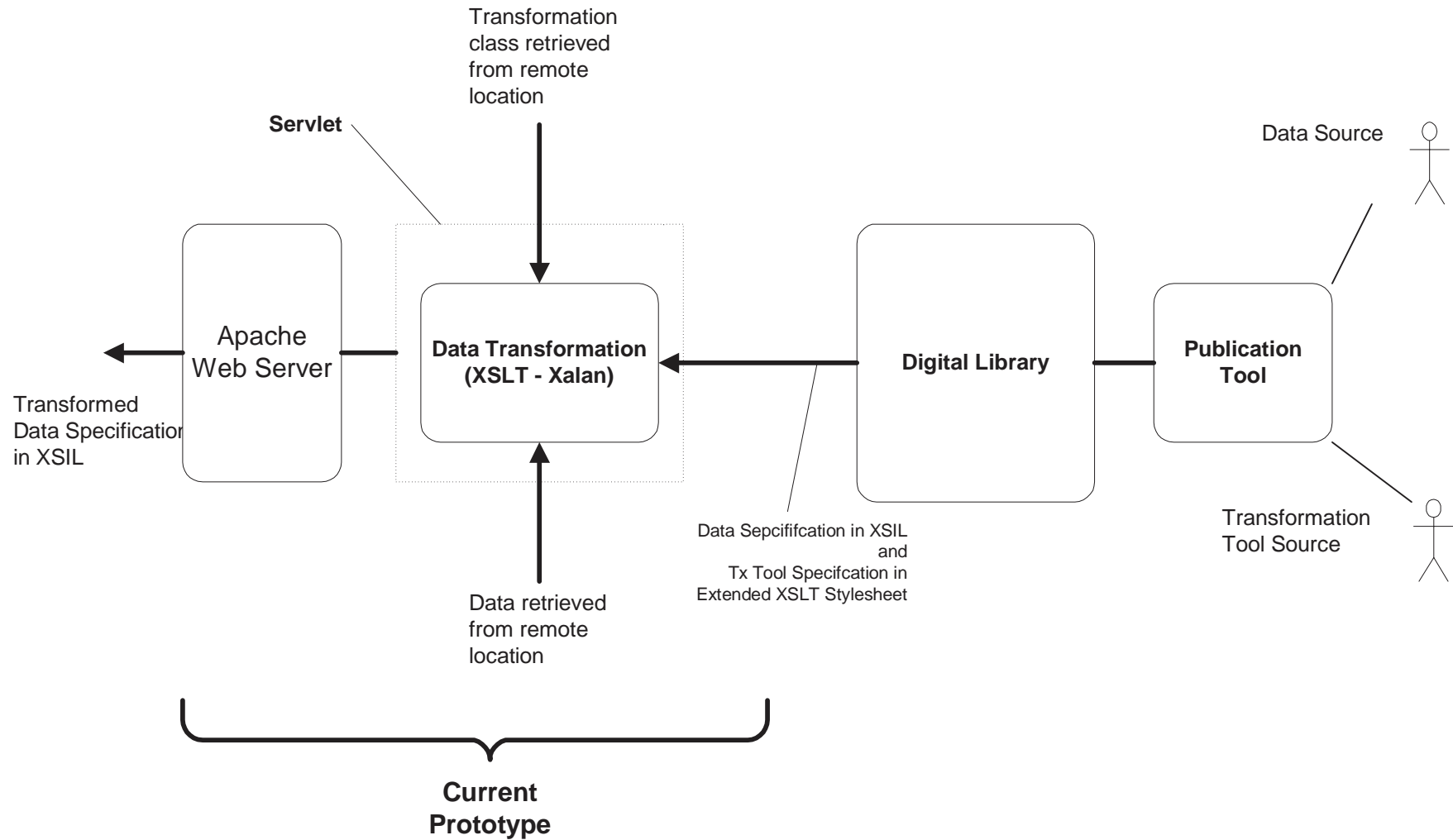
Background

Digital Library -- Software Architecture



XML Scientific Data Management and Processing Facility

Facility Architecture



Extending XSLT

- One major problem faced when using XSLT is its limited functionality, especially in doing scientific data manipulations
- Solution: Use its extension mechanism to define new functions that include any scientific transformation logic

Extending XSLT with Java

Three Step Process for associating extension functions with Java Classes

Step 1. Declare an extra namespace for the extension as well as an extension-element-prefix:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="1.0" xmlns:lxslt="http://xml.apache.org/xslt"  
  xmlns:myxslt="ext" extension-element-prefixes="myxslt" >
```

Extending XSLT with Java

Step 2. Declare the new functions and associate them with Java classes:

```
<lxslt:component prefix="myxslt" functions="polar3">  
<lxslt:script lang="javaclass" src = "XSLTExtensions"/>  
</lxslt:component>
```

Extending XSLT with Java

Step 3. Call the extension functions in the style sheet:

```
<xsl:template match="Stream">
    <xsl:variable name="stream" select="."/>
    <p><xsl:value-of select = "myxslt:polar3(string(@Delimiter),
string($stream))"/ > </p>
</xsl:template>
```

Extending Xalan

- Xalan does not support remote Java classes that have been associated with external functions
- Solution: Extend the Xalan-Java processor to handle the extension function calls specified via a URL by modifying the `ExtensionFunctionHandler.java` in Xalan package `org.apache.xalan.xpath`

Modified Style Sheet

Source changed to include URL

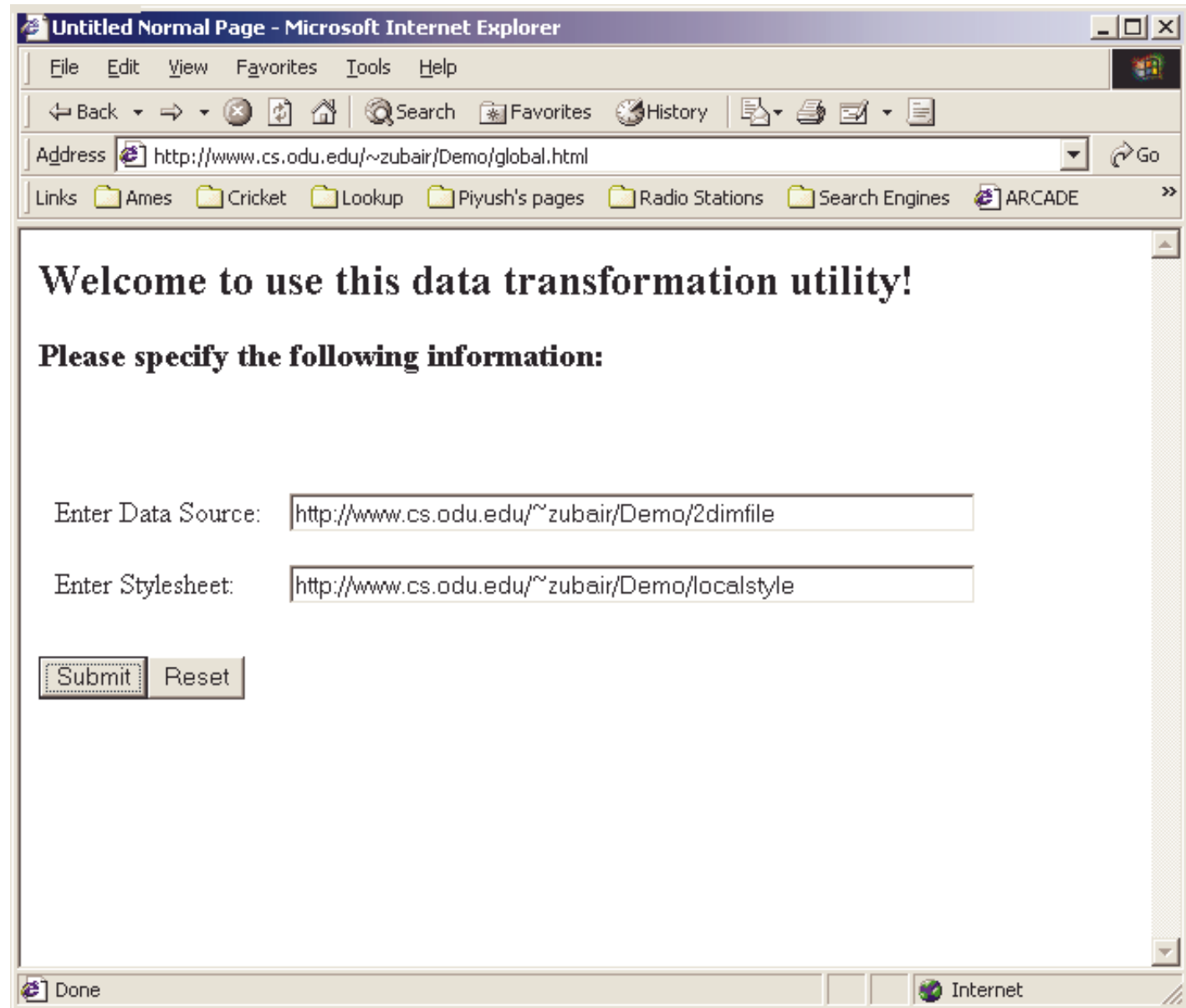
```
<lxslt:component prefix="myxslt" functions="polar3">  
  <lxslt:script lang="javaclass"  
    src="http://www.cs.odu.edu/~tan_d/xslt/XSLTExtensions.class"/ >  
</lxslt:component>
```

Demo

- Demo 1
 - Local data and local transformation class
- Demo 2
 - URL specified data and URL specified transformation class

Demo 1:

Local
Data
&
Local
Transformation
class



The screenshot shows a Microsoft Internet Explorer window titled 'Untitled Normal Page - Microsoft Internet Explorer'. The address bar displays 'http://www.cs.odu.edu/~zubair/Demo/global.html'. The page content includes a welcome message and a form for specifying data source and stylesheet information.

Welcome to use this data transformation utility!

Please specify the following information:

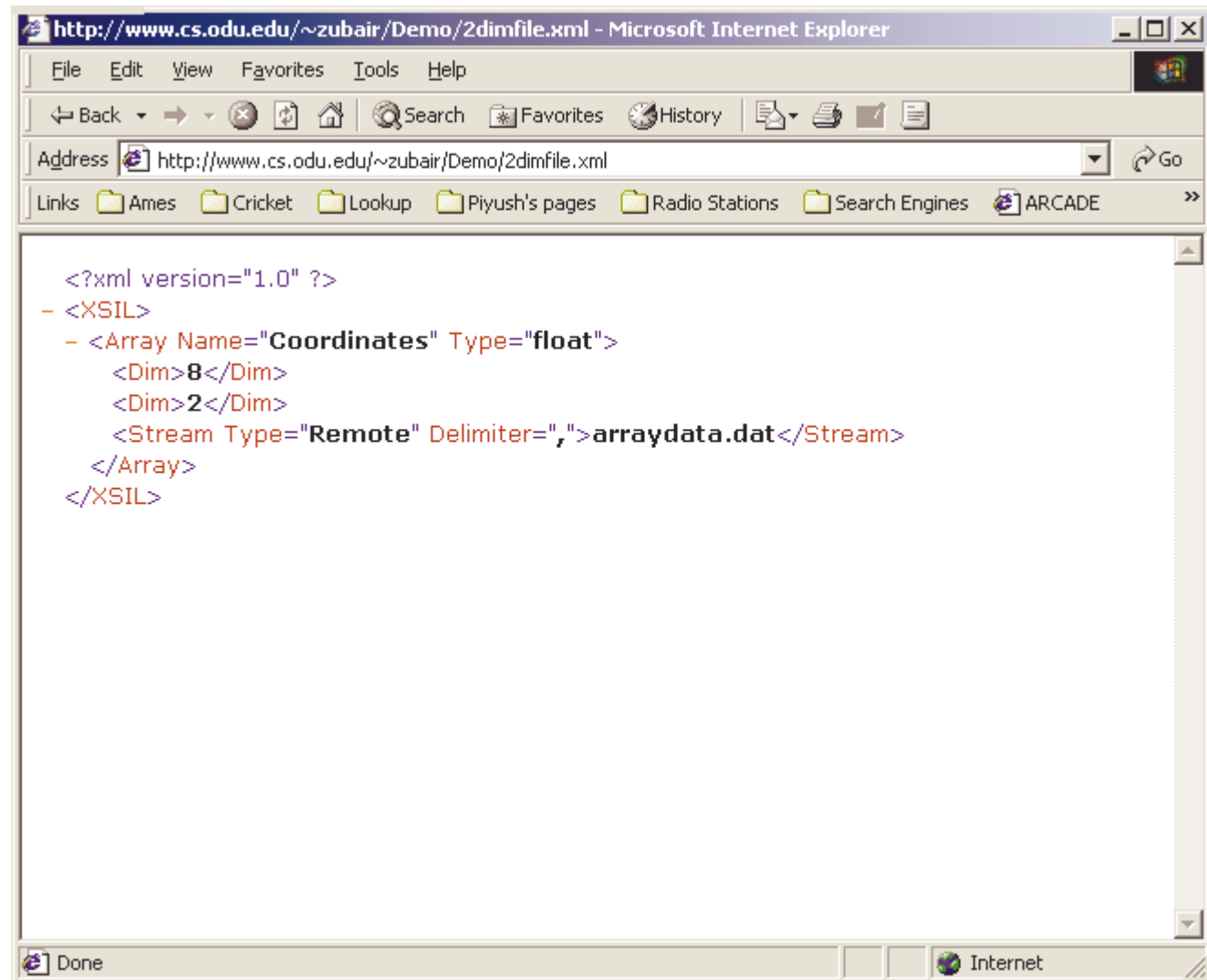
Enter Data Source:

Enter Stylesheet:

The status bar at the bottom shows 'Done' and 'Internet'.

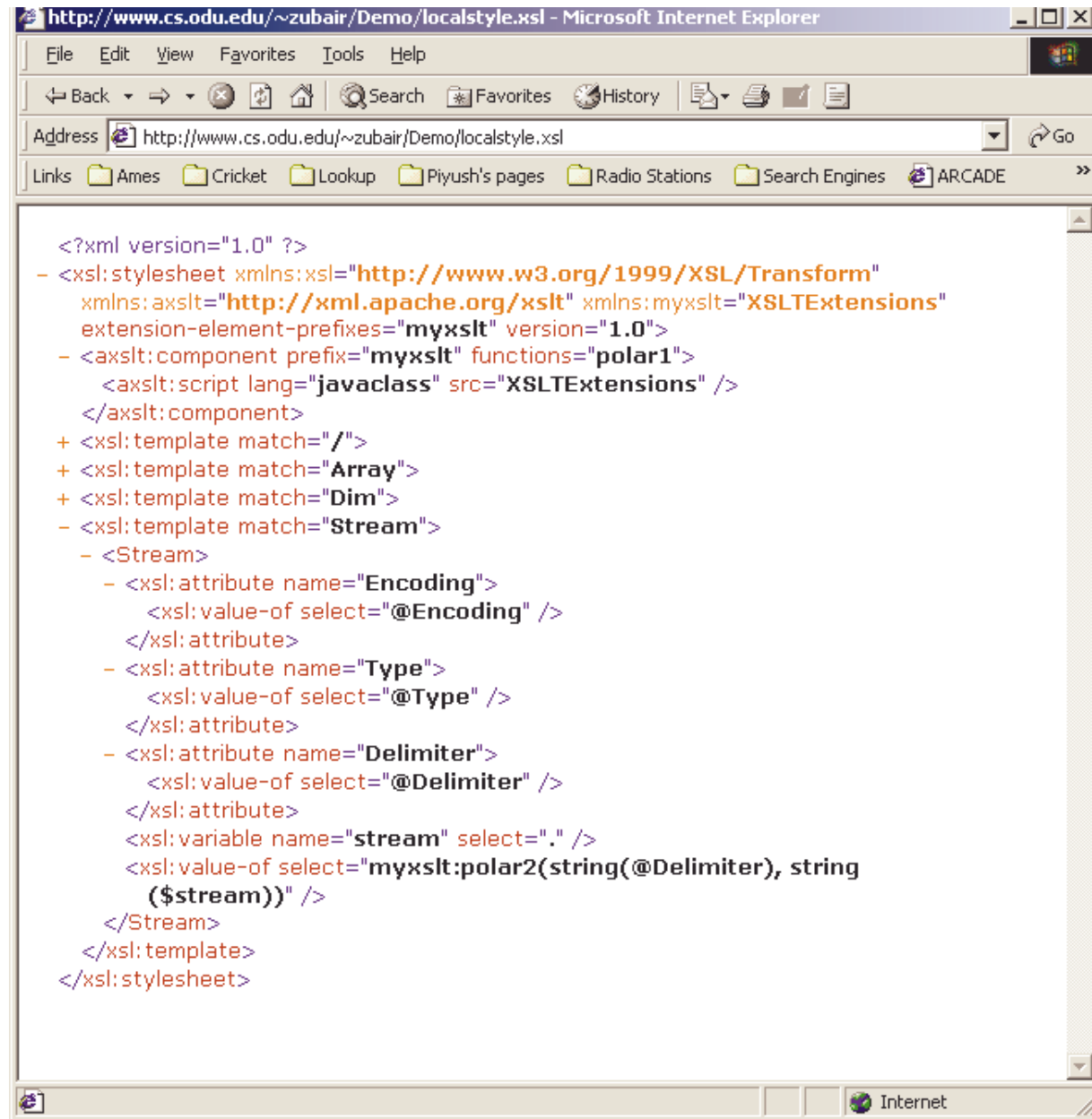
Demo 1:

Local
Data
File



Demo 1:

Local Transformation class



The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://www.cs.odu.edu/~zubair/Demo/localstyle.xml`. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The toolbar contains buttons for Back, Forward, Stop, Home, Search, Favorites, History, and Print. The Address bar shows the URL and a Go button. The Links bar lists several folders: Ames, Cricket, Lookup, Piyush's pages, Radio Stations, Search Engines, and ARCADE. The main content area displays an XSLT stylesheet with the following code:

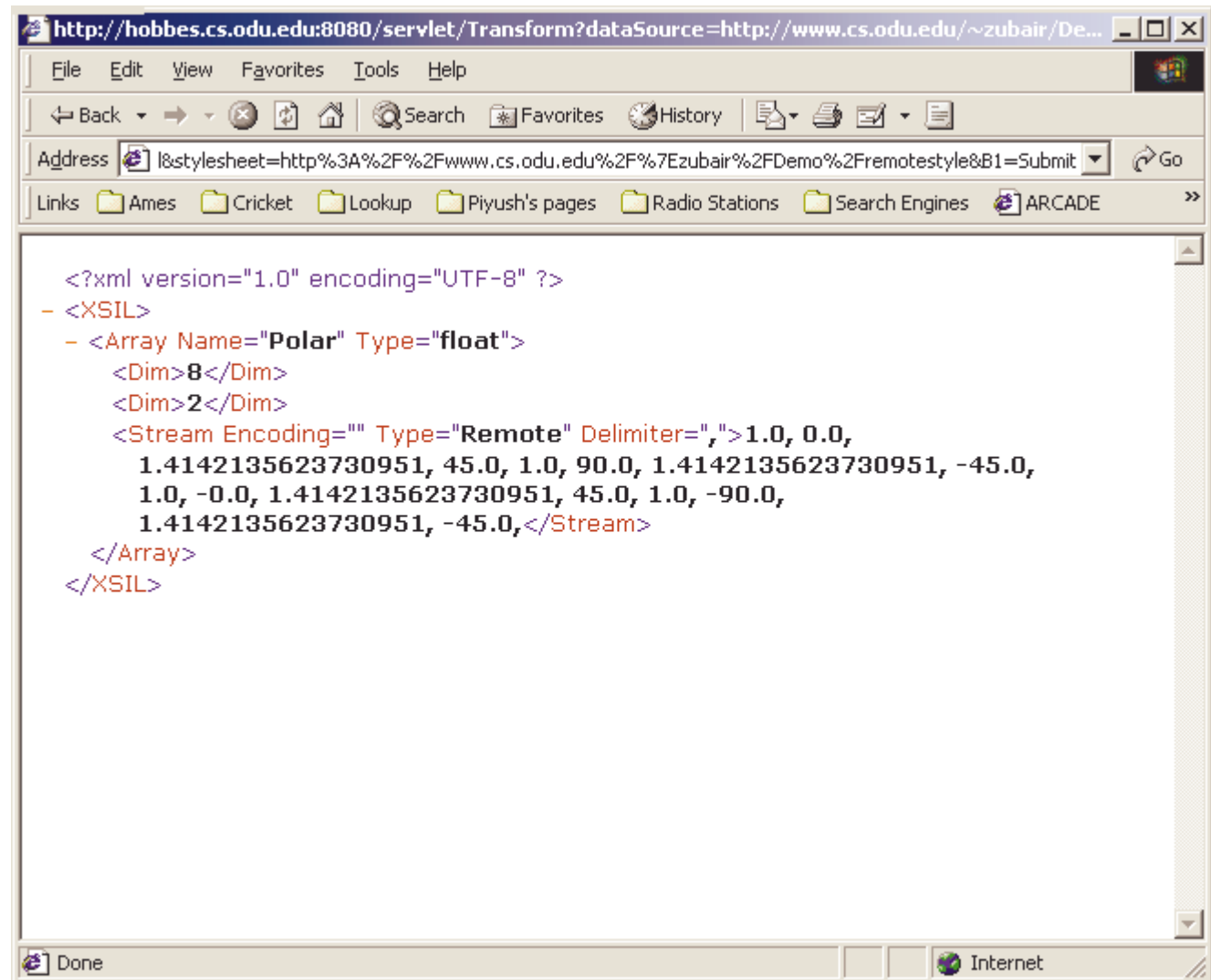
```
<?xml version="1.0" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:axslt="http://xml.apache.org/xslt" xmlns:myxslt="XSLTExtensions"
  extension-element-prefixes="myxslt" version="1.0">
- <axslt:component prefix="myxslt" functions="polar1">
  <axslt:script lang="javaclass" src="XSLTExtensions" />
</axslt:component>
+ <xsl:template match="/">
+ <xsl:template match="Array">
+ <xsl:template match="Dim">
- <xsl:template match="Stream">
- <Stream>
  - <xsl:attribute name="Encoding">
    <xsl:value-of select="@Encoding" />
  </xsl:attribute>
  - <xsl:attribute name="Type">
    <xsl:value-of select="@Type" />
  </xsl:attribute>
  - <xsl:attribute name="Delimiter">
    <xsl:value-of select="@Delimiter" />
  </xsl:attribute>
  <xsl:variable name="stream" select="." />
  <xsl:value-of select="myxslt:polar2(string(@Delimiter), string
    ($stream))" />
  </Stream>
</xsl:template>
</xsl:stylesheet>
```

The status bar at the bottom shows the Internet icon and the text "Internet".

Demo 1

Output :

Local
Data &
Local
Transfor-
-mation
class



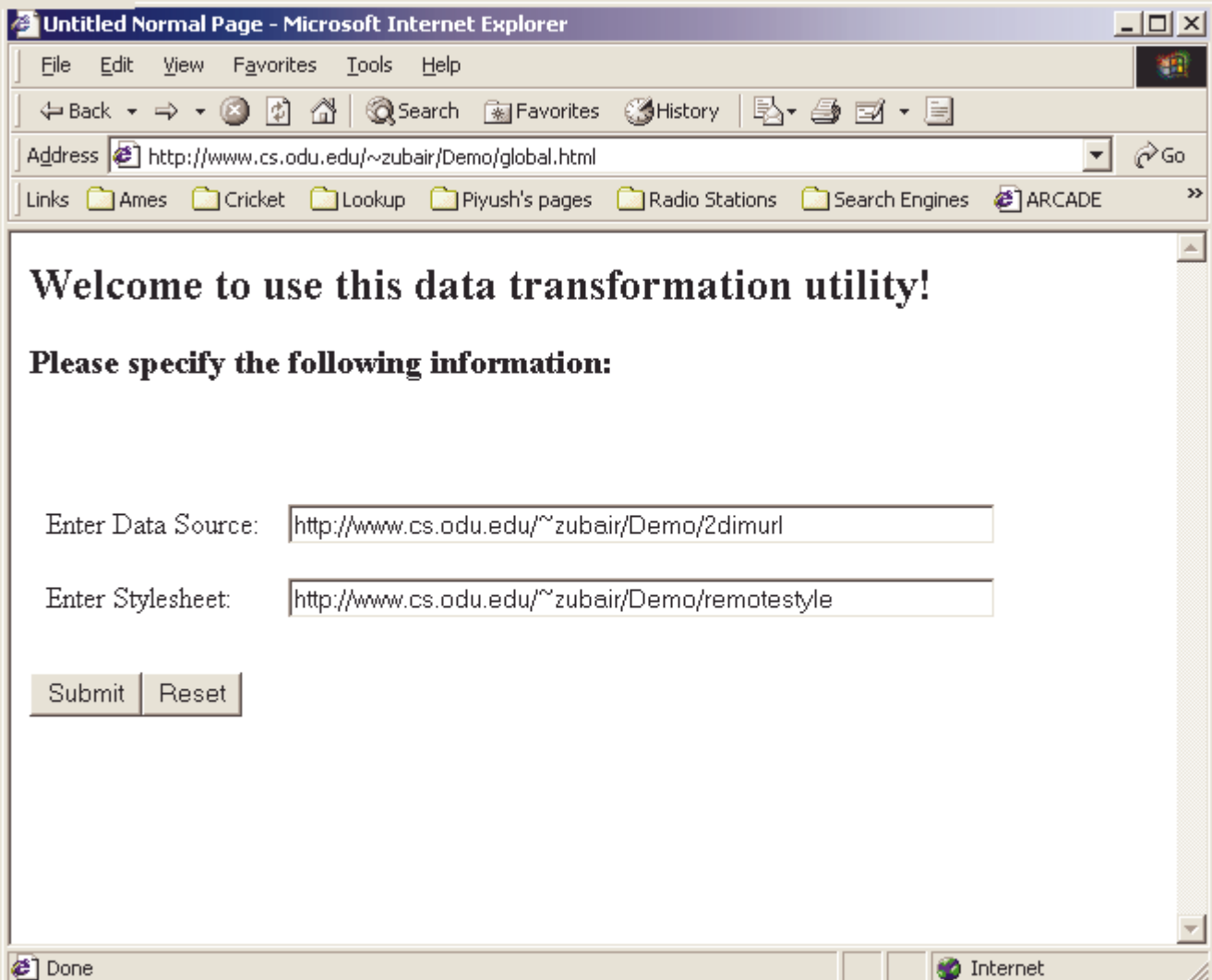
The screenshot shows a web browser window with the address bar displaying the URL: `http://hobbes.cs.odu.edu:8080/servlet/Transform?dataSource=http://www.cs.odu.edu/~zubair/De...`. The browser's address bar also shows a search path: `l&stylesheet=http%3A%2F%2Fwww.cs.odu.edu%2F%7Ezubair%2FDemo%2Fremotestyle&B1=Submit`. The main content area displays XML output from a servlet, which is an XSIL document. The XML structure is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <XSIL>
  - <Array Name="Polar" Type="float">
    <Dim>8</Dim>
    <Dim>2</Dim>
    <Stream Encoding="" Type="Remote" Delimiter=",">1.0, 0.0,
      1.4142135623730951, 45.0, 1.0, 90.0, 1.4142135623730951, -45.0,
      1.0, -0.0, 1.4142135623730951, 45.0, 1.0, -90.0,
      1.4142135623730951, -45.0,</Stream>
    </Array>
  </XSIL>
```

The status bar at the bottom of the browser window shows "Done" and "Internet".

Demo 2:

Remote Data & Remote Transfor- -mation class



The screenshot shows a Microsoft Internet Explorer window titled "Untitled Normal Page - Microsoft Internet Explorer". The address bar displays "http://www.cs.odu.edu/~zubair/Demo/global.html". The page content includes a welcome message, a request for user input, and two text fields for "Enter Data Source" and "Enter Stylesheet", both containing URLs. Below the fields are "Submit" and "Reset" buttons. The status bar at the bottom shows "Done" and "Internet".

Welcome to use this data transformation utility!

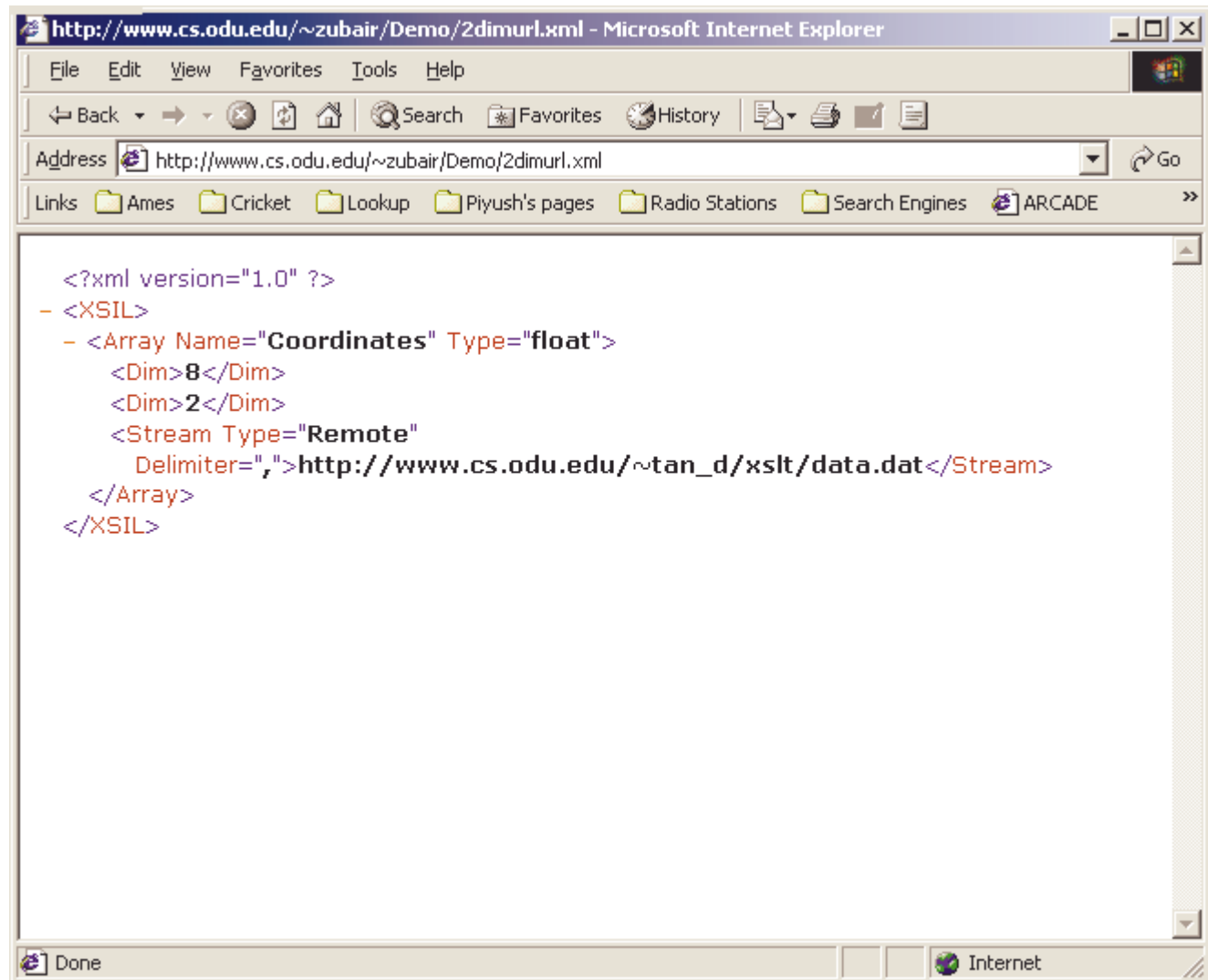
Please specify the following information:

Enter Data Source:

Enter Stylesheet:

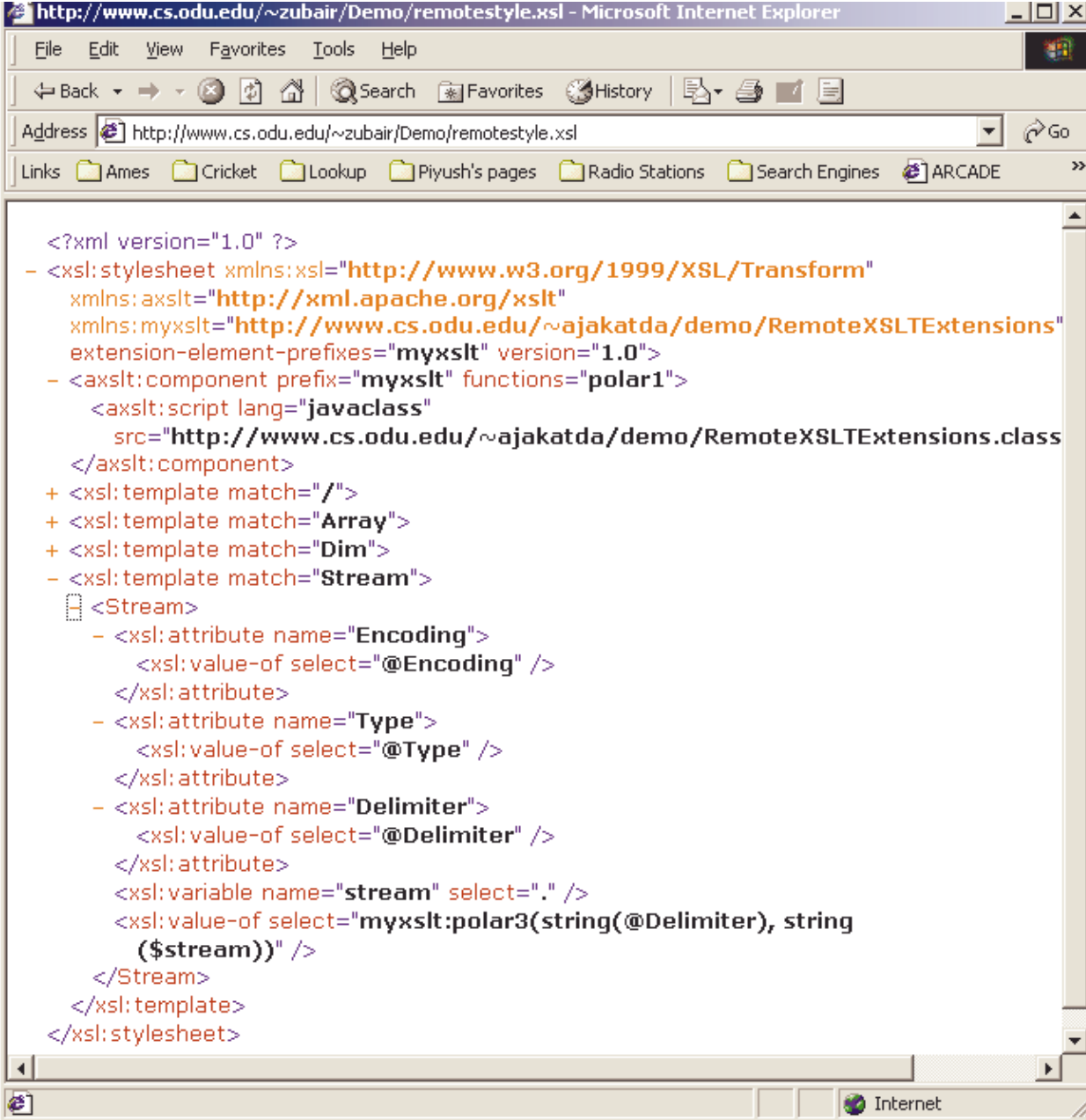
Demo 2:

Remote
Data
File



Demo 2:

Remote Transformation class



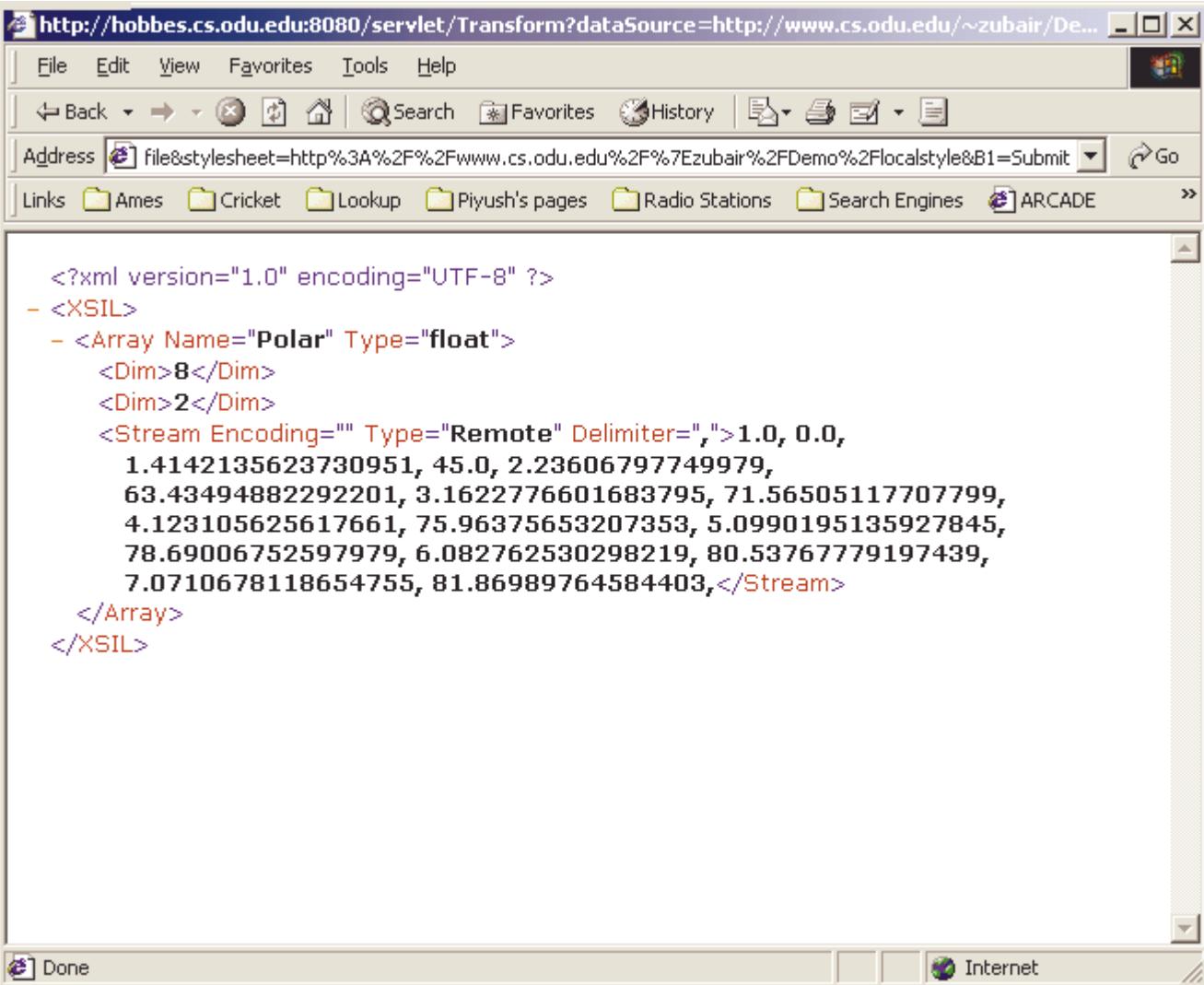
The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://www.cs.odu.edu/~zubair/Demo/remotestyle.xsl`. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The address bar has a 'Go' button. Below the address bar is a 'Links' bar with several folder icons and names: Ames, Cricket, Lookup, Piyush's pages, Radio Stations, Search Engines, and ARCADE. The main content area displays an XSLT stylesheet with the following code:

```
<?xml version="1.0" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:axslt="http://xml.apache.org/xslt"
  xmlns:myxslt="http://www.cs.odu.edu/~ajakatda/demo/RemoteXSLTExtensions"
  extension-element-prefixes="myxslt" version="1.0">
- <axslt:component prefix="myxslt" functions="polar1">
  <axslt:script lang="javaclass"
    src="http://www.cs.odu.edu/~ajakatda/demo/RemoteXSLTExtensions.class"
  </axslt:component>
+ <xsl:template match="/">
+ <xsl:template match="Array">
+ <xsl:template match="Dim">
- <xsl:template match="Stream">
  <Stream>
    - <xsl:attribute name="Encoding">
      <xsl:value-of select="@Encoding" />
    </xsl:attribute>
    - <xsl:attribute name="Type">
      <xsl:value-of select="@Type" />
    </xsl:attribute>
    - <xsl:attribute name="Delimiter">
      <xsl:value-of select="@Delimiter" />
    </xsl:attribute>
    <xsl:variable name="stream" select="." />
    <xsl:value-of select="myxslt:polar3(string(@Delimiter), string
      ($stream))" />
    </Stream>
  </xsl:template>
</xsl:stylesheet>
```

The status bar at the bottom of the browser window shows the Internet icon and the text 'Internet'.

Demo 2 *Output :*

Remote
Data &
Remote
Transfor-
-mation
class



The screenshot shows a web browser window with the address bar containing the URL: `http://hobbes.cs.odu.edu:8080/servlet/Transform?dataSource=http://www.cs.odu.edu/~zubair/De...`. The browser's address bar also shows a file path: `file&stylesheet=http%3A%2F%2Fwww.cs.odu.edu%2F%7Ezubair%2FDemo%2Flocalstyle&B1=Submit`. The main content area displays XML output from a servlet. The XML is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <XSIL>
- <Array Name="Polar" Type="float">
  <Dim>8</Dim>
  <Dim>2</Dim>
  <Stream Encoding="" Type="Remote" Delimiter=",">1.0, 0.0,
    1.4142135623730951, 45.0, 2.23606797749979,
    63.43494882292201, 3.1622776601683795, 71.56505117707799,
    4.123105625617661, 75.96375653207353, 5.0990195135927845,
    78.69006752597979, 6.082762530298219, 80.53767779197439,
    7.0710678118654755, 81.86989764584403,</Stream>
</Array>
</XSIL>
```

The browser's status bar at the bottom shows "Done" and "Internet".

Future Work

- Support extension functions that can store transformed data on local disk
- Provide support for remote execution of transformation function
- Metadata specification for tools and scientific data
- Support for publishing and discovery tools via a Digital Library system
- Develop API so as to integrate into Arcade and other frameworks
- Performance study